Using SDL for Prototyping Low Rate Wireless Personal Area Networks

David Cypher

Advanced Networking Technologies Division (ANTD)/ Information Technology Laboratory (ITL)/ National Institute of Standards and Technology (NIST), Stop 8920, 100 Bureau Drive, Gatihersburg, MD 20899 U.S.A.

Abstract. This paper describes the use of the specification and description language (SDL) as a prototyping tool for a developing low rate wireless personal area network protocol. A simplification of the guidelines for protocol formalization using SDL is used to create a behavioral model. This behavior model is then hand-tested for protocol behavior using software tools that permit simulation of the SDL model. Simulation of peer SDL models is done next to ensure interoperability. Finally a formal validation tool is used on the resultant SDL models.

1 Introduction

As the need to generate wireless communication standards at a break-neck pace intensifies, it becomes more critical to quickly simulate, validate, and test the protocols contained in the developing draft standards. A method for doing so is rapid prototyping using the specification and description language (SDL) [1] and related tools. This paper describes the application of the SDL as a rapid prototyping tool for the creation of a behavior model of protocols for a low rate wireless personal area network. It also describes the various protocol testing and modifications necessary to complete the behavior SDL model.

Since the time and resources necessary for the generation of a complete and formal description of the low rate wireless personal area network protocol (WPAN) using the SDL do not exist, a best effort behavior, or logical, model using SDL is a practical solution. Therefore the goal of this work is not to define a formal description of the protocol(s) using SDL, but rather to provide some assurance that the protocol(s) under development are complete and perform the functions that were intended. That is; we want to avoid what is stated in clause 5.3.1 of [2], "Specifying a protocol without first evaluating what it is intended to achieve and what constraints are to be applied to, it will almost certainly end in a poor specification."

With this as the stated goal, the guidelines provided in [3] are used as a starting point. The guideline contains a three-step process: structure, behavior, and data for the use of formal SDL as a descriptive tool. For the most part all of the steps in the structure and behavior steps are followed. The various step requiring comments within the SDL model are not followed, since it is expected that the SDL model will change significantly from one iteration to the next and the time needed to provide comments could more appropriately be used elsewhere. The data step is almost entirely ignored, since the goal is not to describe a formal model using SDL. For this reason there is no use of the abstract syntax notation number 1 (ASN.1). The built in SDL data types are use, especially the literal type (enumeration).

2 Low Rate Wireless Personal Area Network

The Task Group 4 (TG4) of the Wireless Personal Area Network (WPAN) Working Group (802.15) of the Institute of Electrical and Electronics Engineers (IEEE) 802 Local Area Network (LAN)/ Metropolitan Area Network (MAN) Standards Committee is developing a draft standard for a low rate WPAN. This development process went from basically a requirements document [4] (about 120 pages) in its first working group letter ballot to a limited protocol specification [9] (over 200 pages, not including the annex containing the SDLs) in its first sponsor ballot. This process took a period of a little over 15 months. The process is all but over. The content of each draft changed significantly, which led to the SDLs contained in the annex to be out-of-sync with the text description in the draft.

The initial goals and applications of this low rate WPAN are described in [10], as well as the purpose and scope of the draft standard.

3 SDL Creation

The SDL models were created by selectively following the steps listed in the Guidelines for the Use of Formal SDL as a Descriptive Tool [2]. Since the goal of the guidelines (i.e., a formal description in SDL) and the goal of this work (i.e., best effort behavior model) are not the same, not all of the steps are followed. The selected steps are listed and either show how the step was applied or describe why the step was not applied. The guidelines describe three main steps. Each of these three steps is further subdivided into explicit instructions.

Using this modified process for SDL creation, six skeleton SDL models were created. Each SDL model is as close to the draft text as time and resources would permit. Some models are more complete than others. The goal was to evaluate the completeness and correctness of the functional descriptions and operations contained within the current draft before the end of the commenting and balloting period, which was either 10 or 40 days. Reuse of an SDL model was quickly discounted in some cases when the draft changed so significantly that it was deeded better to start completely over. While under short time periods, reuse was the only possibility. This is the major reason for not adding comments or documentation in the SDL model. Comments are added as a note where assumptions are made or errors are found.

3.1 The Structure Step

The structure step contains eight sub-steps. Basically this is the highest level design for a protocol and the beginning for using SDL. Each of the sub-steps is now listed and described.

S.1 Boundary and Environment of the System. Identifying the boundaries between the system to be described and its environment, finding a suitable name for the system, and drawing an SDL system diagram were simply a matter of applying the IEEE 802 structure and naming conventions. Thus the boundaries were defined by the service access points (SAP), the system name was that of the task group (i.e., TG4_MAC_PHY). See Figure 1 for the SDL system.

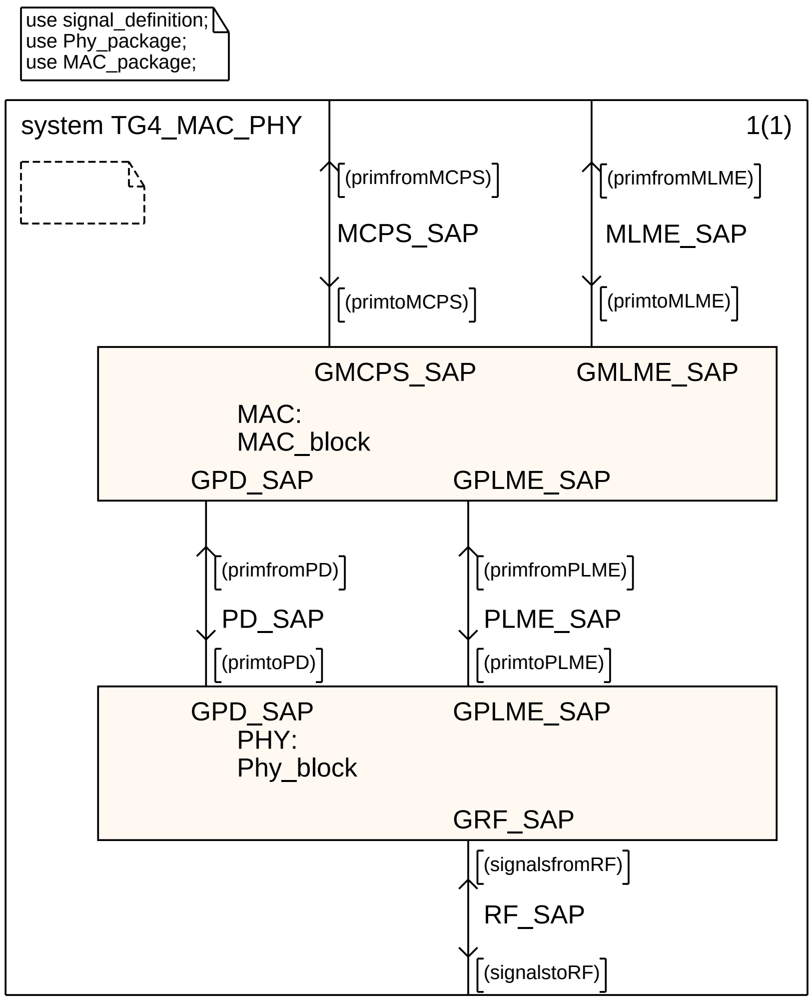

Figure 1 SDL system for the PHY and MAC Block

S.2 Discrete System Communications. There was no need to identify the entities outside of the system or to identify the information flow, since these were already described in the draft text. Modeling the messages by signals was a straight forward translation from text to SDL. Stating the relationship between each signal and entities external to the system and stating the purpose for each signal in a comment with the signal definition were not done because the likelihood that the signal name and function would remain unchanged was almost nil. These were also considered documentation, which are not considered as important as other steps in the creation process, especially when there is no control over revisions. Placing related signals for one entity in one direction into a signallist is followed because it confines the addition or deletion of signals to one location. The signals and signallists are defined outside of the SDL system model in an SDL package. This SDL package is made available to the SDL system via the USE feature.

S.3 System Parts. Two main parts are identified within the system: the MAC sublayer and the physical layer. Both parts are drawn as blocks, actually block types, and use the same names. However the adding of the description for each block was ignored. Both block types are part of separate SDL packages. Use of these SDL packages simplify the reusability across the various SDL systems that are used during testing, simulating, and validating.

S.4 Communication Paths between Parts. Identifying the channels needed between blocks and the boundary of the diagram, and identifying the direction of communications are defined by the text description and by following the primitive descriptions. Associating a signallist and a name to these channels are straight-forward. The channels have an acronym for the type with a suffix of _SAP. The acronyms are MAC common part sublayer (MCPS), MAC sublayer management entity (MLME), Physical data (PD), Physical layer management entity (PLME), and radio frequency (RF). The signallist name was chosen to represent the direction (to or from) and type (MCPS, MLME, PD, PLME, or RF) of signals used. These signallists and channel names are shown in Figure 1.

S.5 Associating Signals to Communication Paths. Naming, defining, and associating the signals to the appropriate signallist followed the text description in the draft. However for each new draft version this adding, removing, or renaming of the signals became a management issue. Sometimes it was determined better to remove all the signals from the list and start again. At other times it required a name change to the signallist, but not the signals. The SDL editing tool's find and replace feature is extremely useful for this.

S.6 Information Hiding and Sub Structuring. All blocks were determined to contain only processes. Further sub-dividing either of the main blocks into other blocks created more of a problem when it came to information distribution. The draft text described two entities within each of the major parts (i.e., MAC sublayer and PHY layer). These two parts are the management entity and the data entity. However the description of the two does not lend itself to divide easily their function into two separate blocks. This was done in one of the versions, but the number of and the need for internal signals and the passing of duplicate information became too cumbersome. Thus there was no need to recursively use this step.

S.7 Block Constituents. Much like all of the other steps, this was already defined in the draft, so translating the draft text into SDL was the only step. Each process was defined to be one.

S.8 Logical Signals in a Block. Identifying and defining logical signals within a block have come and gone between different interactions. In older versions of the SDL model logical signals were easily identified by prefixing the name with an "i". The "i" prefix indicated that the signal was an internal block signal needed for proper operations, but not defined in the draft. These signals were used mainly within the MAC sublayer block between the management and the data blocks. However this sub-division of the MAC sublayer block into two other blocks no longer exists in the latest SDL model. No procedures were imported or exported. Procedures are defined where they are used.

3.2 The Behavior Step

The behavior step contains four sub-steps. This is the area of most concern for the goals of the best effort behavior model using the SDL. All of these steps are followed, except for the comments.

B.1 Set of Signals to a Process. Since the draft text had the signals already defined, identifying signals to a process was already done.

B.2 Skeleton Process. Simple message sequence charts (MSC) were included as part of the draft text, so this work was already done. More extensive MSCs are now part of the draft text. The MSCs in the latest draft show more detailed operations than in previous drafts, especially the interactions between the MAC sublayer and the physical layer. No MSC are created as part of the SDL prototyping, except when automatically created and viewed as part of the testing, simulating, or validating process.

Producing a skeleton process is the most intensive and difficult part for the creation of the SDL models. Building a tree of states is required with every new draft text. The draft text does not define any explicit states for the MAC sublayer process, so this is open for interpretation. In the latest model there are eleven states and sixteen transition states. The eleven states are stable states (i.e., the state will not change unless forced into another state).. The transition states are part of a sequence that is started and will be in that state only temporarily until the sequence is completed. The draft text does however provide some guidance for the states at the physical layer. Currently the physical layer has at least three states (receiver on, transmitter on, and transmitter and receiver off). Two more states have been proposed from time to time, but neither have been included. The process tree was built by defining a branch for each input signal. Adding a new state, if the behavior for the input signal was different, was like a runaway train. The number of states quickly expanded to an unusable number of states. Many attempts were made to reduce this number, but none succeeded due to the various, but only slightly different conditions. This necessitated the addition of more states. The numerous timers defined in the text were added to the SDL models as a last step, rather than at this sub-step. Most of the timers are for recovery from a loss of the communication channel between peer entities. The assumption in this first round of prototyping is that the communications channel is never lost. Phased another way if the protocols' behavior does not work when the communication channel is good (no signal loss), then why bother testing the protocol on a bad communication channel (where signals are lost). This assumption must be revisited later in the process, since the communication channel is a wireless one.

Drawing the graph as a process diagram can take one of two forms. Either the process can fix the state and then exhaust the input signals for that state before proceeding to the next state, or the process can fix the input signal and exhaust all of the states before proceeding to the next input signal. Both methods are used, which leads to some confusion when trying to find a state and signal combination. The indecision on using a single method is based on the level of completeness of the draft. If a final SDL model would be created, then only one method would be used. However, as long as the SDL is used for prototyping, time will not be spent to convert to only one style, which is considered editorial clean up. The physical layer process lists the state first and all its input signals before listing the next state. The MAC sublayer process follows no single method.

Determining whether the process has two or more disjoint sets of interfaces for different behavior is another place of real confusion. The draft text is not well defined in this area and so as the SDL model is created, suggestions and assumptions are made, which need to be made into comments and submitted to the writers of the draft text for clarification.

B.3 Informal Processes. The identification of combinations of use cases is used to simulate the SDL model for proper operations. It is also used to structure the ordering of the states and signal inputs within the process. Identifying what information the process stores and using this information to define the actions of each process was mostly a matter of following the draft text, but many omissions exists, to which suggestions and assumptions are made. Adding tasks and procedures is done on a case by case basis. First a task is commented on in the branch. The comment is later removed and replaced with the actual task or procedure, when the operation of the branch is necessary for execution to advance the state of the process during simulations. Decisions, too, are added first as comments and later expanded, when it is determined that the decision is important for the progression of the state of the process. Procedure names provide a description of its intended function. This is important since other commenting steps are not used. Procedures when first defined only contain only the start and return symbols. Sometimes this is done simply for lack of time, while at other times it is done to mark procedures that are best left for implementations. For example randomly picking a number or calculating an cyclic redundancy check (CRC).

B.4 Complete Processes. For each state the signal is processed. No saving of signals is permitted. At this stage in the SDL creation all the inputs are known, but the states and transitions are not. At this point some of the transitions are marked as to be determined because the draft text considers only the most common and usual situations. Some situations are determined by a logical guess. On those that have multiple possibilities comments are submitted to the writers of the draft text for clarification.

The analysis of the individual procedures, process, and block is performed using the simulation tool that permits a running of the SDL model, so that a hand or eye inspection is possible for the behavior. Later after passing the hand or eye inspection an automatic validation tool is used to evaluate the SDL models. It is these two sub-steps that are the most important to achieving the goal for rapid prototyping.

3.3 The Data Step

The data step contains nine sub-steps. Most of these nine data sub-steps are not followed. It is at this step that most others would use ASN.1 for a formal SDL model. However for the purposes of prototyping the built in SDL data features are sufficient.

D.1 Signal Parameters. The draft text defines the signals along with their parameters. To accelerate this step predefined sorts are used almost exclusively, even when they do not match the sort defined in the draft text. For example the CRC is not defined as a 16-bit integer value as defined in the draft, but rather it is defined in SDL as a boolean sort. The reasons for doing this are defining a calculation in SDL is not as easy as using “C’ code and for the purposes of testing behavior either the CRC is correct (True) or it is wrong (False). There is no attempt to correct bit errors or to determine which bits are in error, only to detect them using the CRC, thus the result is always a binary decision. Most non-predefined sorts are literal lists or are listed as synonyms. Examples of the literal lists are the cause or reason codes. Their name is used, rather than their assigned value. This is done to reduce the amount to time to update the actual values as they change from one draft version to the next, while the names are more stable. Synonyms are used for expansion purposes. For example the IEEE address is defined to be 64 bits in length, but for the purpose of these SDL models, the need is only for an unique identifier. Thus the IEEE address is defined as a synonym to be an integer. The actual 64 bits are not needed, but it is expected that this model may be used to finish a formalization of the standard.

The signals, defined previously, are expanded to include these parameters.

D.2 Process and Procedure Parameters. No process or procedure parameters are declared. Scoping rules permit the needed variables to be used by a process or procedure to be available and only variables that are needed within the process or procedure are declared.

D.3 Signal Variables. Adding and defining variable parameters are done, but not the stating of the role of each variable in a comment. Commenting is not done, since most of the parameters and variables change from one version of the draft to the next.

D.4 Formal Transitions. The replacement of the informative text in the tasks, procedures, and decisions is done when the transition is deemed important for the next state or testing of a signal and its function. Other variables and synonyms, as well as procedures are added as needed.

D.5 Output and Create Parameters. None exist, since they are not needed.

D.6 Data Signatures. None are present.

D.7 Informal Data Description. Not provided due to time constraints.

D.8 Formal Data Description. Not provided due to time constraints.

D.9 Complete Data Formalization. Not provided, since the goal of this modeling is to test the behavior, not to create a formal description that would include the coding and ranges of values.

4 Testing, Simulation, and Validation

At many of the steps and sub-steps along the way, varying levels of testing, simulating, and validating are exercised. To accomplish these other SDL systems are created. There is no point to fully draw or define an SDL model and never test any of its smaller component parts as the SDL model is being created. Therefore after adding an amount of information to the SDL model, an analysis of the SDL syntax was done. Any errors or warnings are fixed and then more information is added. After enough information is added to simulate the behavior for one action or input signal, the SDL model is once again syntax checked and a simulation object created. Using the simulation graphical user interface for the SDL simulation tool, a signal is tested for correct behavior. This process is done recursively until either all input signals are simulated and behaviors modified, if necessary, or an ambiguity is found that prevents further simulation until the writers of the draft provide clarification. Once the process is visually examined and hand tested to provide the behavior as described in the draft text, a single block and its process is removed from the original system and placed into another system where only that block and its process exists. Figure 2 shows the physical block in a SDL system for testing only the physical block. This SDL system is then validated using the provided validation tool. If the results are considered good, then the block and its process are tested in yet another SDL system. In this new SDL system a single block and its process are peered to a duplicate of itself via a communication channel. This testing and simulation is necessary to determine if the processes are symmetric and function as expected. For the PHY block this is important. The SDL system containing the peered physical blocks is shown in Figure 3. After a successful completion of this, the peer system is validated. This validation usually results in less paths to eye or hand review than the single block validation, since the peering of this block reduces the number of possible paths or transitions.

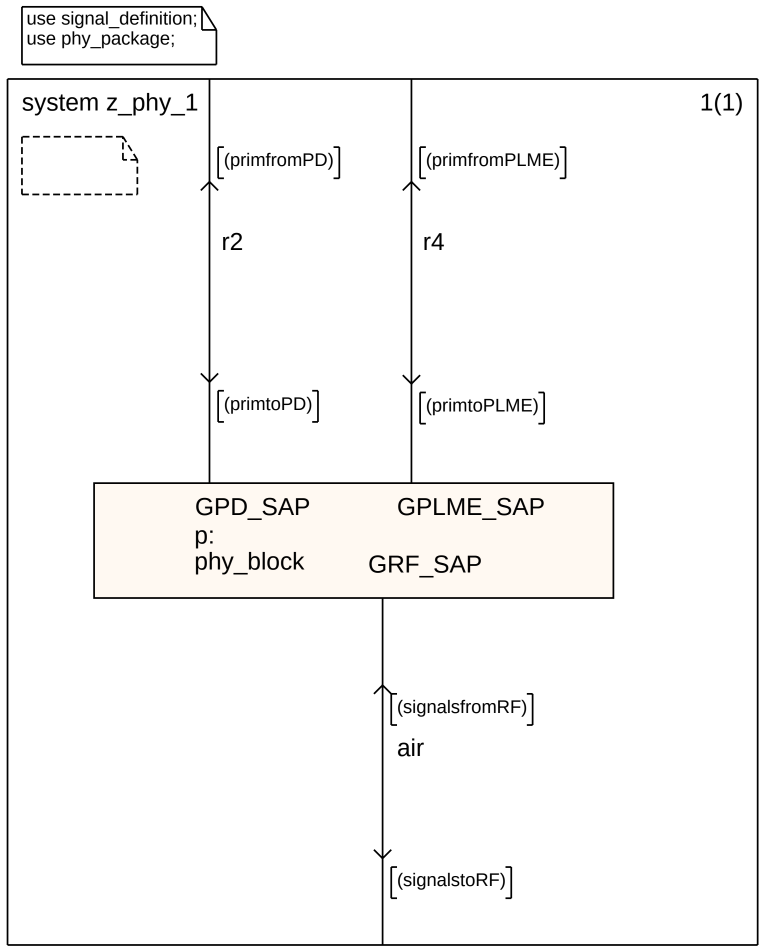

Figure 2 SDL System with a Single (Physical) Block

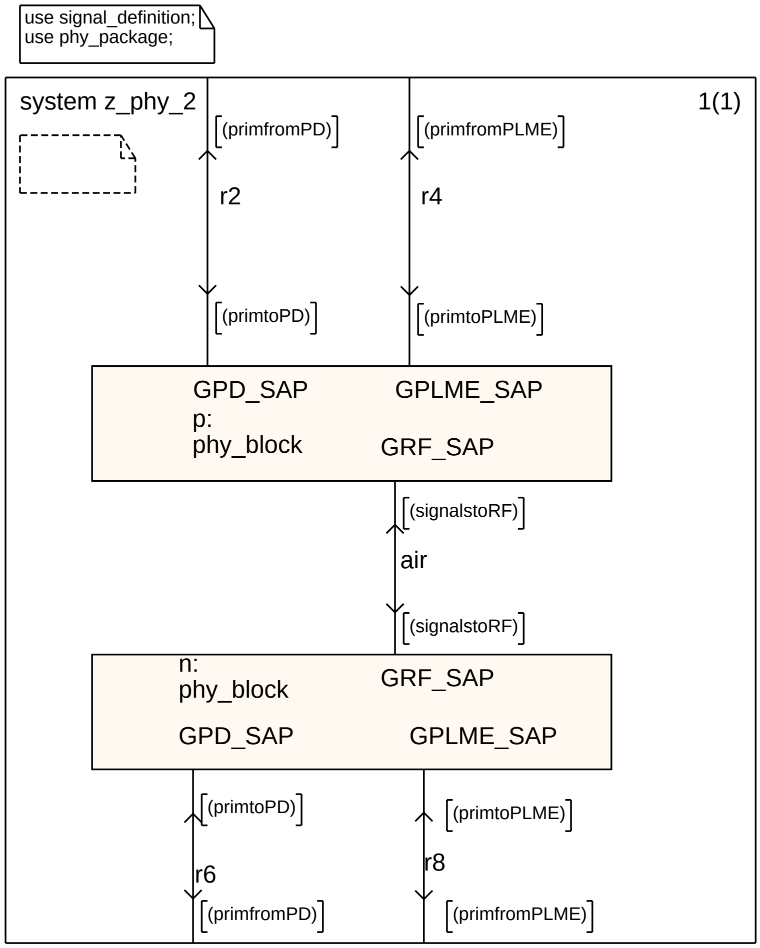

Figure 3 SDL System with Peered Physical Layer Blocks

Both the PHY and MAC blocks are tested for layer interaction. During this testing and simulating, ambiguities and collisions within the draft text are found. Signals are received without enough information contained in their parameter lists to perform the described function. Also sequences of events are found that are required for proper functioning, but yet are not described in the draft text.

After both blocks (i.e., PHY and MAC) complete both types of testing (peer and layer), the entire system is peered. The SDL system for peering the PHY and MAC blocks as a unit is shown in Figure 4. It was at this point that a dead lock was discovered in the model. The draft text was reviewed to ensure that the SDL model was following the behavior described in the draft text. A description of the deadlock was then submitted to the writer's of the draft for modification.

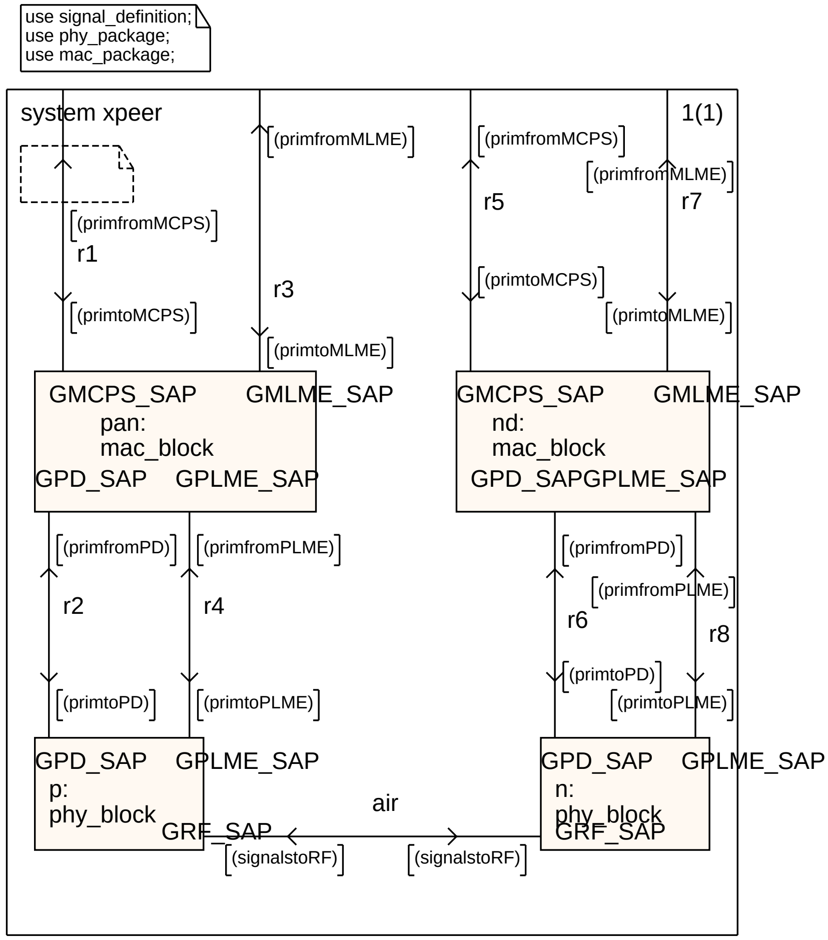

Figure 4 SDL System for Peered PHY and MAC Blocks

5 Conclusions

By selectively using steps from the guideline [2] (mostly by removing all of the comment and most of the data sub-steps), an SDL model can be quickly developed. This rapid development of a model for testing, simulating, and validating of the contained behavior of the developing protocol was useful to the standards developing cycle. This process uncovered various ambiguities, unspecified transitions, and even a deadlock within the draft text. All of these findings were submitted to the writers of the draft prior to the finalization of the draft standard. Thus helping to ensure that at least those errors found now will not exist in the final published standard and that the final standard will not be a poorly described specification.

A new appreciation for the value of using SDL and its ability to be used as a rapid prototyping tool (implementation) to test behavior during the standard's development cycle has gained some level of acceptance. There are still those, however, that insist that any SDL model that does not completely describe the protocol is useless, wrong, or a misuse of the SDL.

These prototypes, or best effort, models using SDL can be made useful to others, such as implementers and testers, by finishing the steps not taken in this application of [2] to produce a complete and formal description. However the standard must be finalized before this effort should be undertaken.

References

[1] ITU-T Recommendation Z.100, Specification and Description Language (SDL)

[2] ETSI EG 202-106 v2.0.0 (2002-07), Methods for Testing and Specification (MTS) Guidelines for the Use of Formal SDL as a Descriptive Tool

[3] ETSI ETR 298 (1996), Methods for Testing and Specification (MTS); Specification of Protocols and Services; Handbook for SDL, ANS.1 and MSC Development

[4] IEEE P802-15-4/D13 (December 2001) Draft Standard for Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)

[5] IEEE P802-15-4/D14 (April 2002) Draft Standard for Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)

[6] IEEE P802-15-4/D15 (June 2002) Draft standard for Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)

[7] IEEE P802-15-4/D16 (July 2002) Draft Standard for Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)

[8] IEEE P802-15-4/D17 (October, 2002) Draft standard for Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)

[9] IEEE P802-15-4/D18 (February, 2003) Draft standard for Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)

[10] Ed Callaway, Paul Gorday, Lance Hester, Jose A. Gutierrez, Marco Naeve, Bob Heile, Venkat Bahl, "Home networking with IEEE 802.15.4: a developing standard for low-rate wireless personal area networks", IEEE Communications Magazine, vol 40 No.8, August 2002,.

[11] Jian Huang, Yan Huang, Ed Callaway, Qicai Shi, and Bob O'Dea, "Simulation of a low-duty-cycle protocol," Proceedings of OPNETWORK 2001, August 27-31, 2001, Washington, D.C.

[12] J.M. Alvarez, M.Diaz, L.M. Llopis, E. Pimentel, J.M. Troya, “SDL and hard real time systems: new design and analysis techniques”, 2nd Workshop on SDL and MSC, Grenoble, France, June, 26-28 2000

[13] M.Bozga, S.Graf, A.Kerbrat, L. Mounier, I. Ober, D. Vincent, “SDL for real time: What is missing ?”, 2nd Workshop on SDL and MSC, Grenoble, France, June, 26-28 2000

[14] A. Olsen, O. Faergemand, B. Moller-Pedersen, R. Reed, J.R.W. Smith, “System Engineering Using SDL-92”, North-Holland, 1997.

[15] Jan Ellsberger, Dieter Hogrefe, Amardeo Sarma, “SDL Formal Object-oriented Languages for Communications Systems”, Prentice Hall, 1997.

[16] Gerard J. Holzman, “Design and Validation of Computer Protocols”, Prentice Hall, 1991.